Assignment 1: Mmm, How Fast It Can Be?

2020 Fall EECS205002 Linear Algebra

Due: 2020/10/19

Matrix matrix multiplication (Mmm) is one of the fundamental operations in numerical algorithms. Let A, B, and C be three $n \times n$ matrices, whose elements are denoted as $a_{i,j}$, $b_{i,j}$ and $c_{i,j}$ respectively for $1 \leq i, j \leq n$. Based on the definition of matrix multiplication $C = AB$,

$$c_{i,j} = \sum_{k=1}^{n} a_{i,k} b_{k,j}, \tag{1}$$

we have the basic matrix multiplication algorithm, as shown below.

Algorithm 1 Matrix-Matrix Multiplication

```
for i = 1 to n do
  for j = 1 to n do
    C[i][j] = 0
    for j = 1 to n do
      C[i][j] = C[i][j] + A[i][k]B[k][j]
    end for
  end for
end for
```

This is a three loop algorithm. You can calculate the number of additions is n^3 and the number of multiplication is also n^3.

Now we consider another type of algorithms, called *block matrix multiplication*. It is based on the property of block matrices. Let

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,M} \\ A_{2,1} & A_{2,2} & \cdots & A_{2,M} \\ \vdots & \vdots & \vdots & \vdots \\ A_{N,1} & A_{N,2} & \cdots & A_{N,M} \end{bmatrix}, B = \begin{bmatrix} B_{1,1} & B_{1,2} & \cdots & B_{1,N} \\ B_{2,1} & B_{2,2} & \cdots & B_{2,N} \\ \vdots & \vdots & \vdots & \vdots \\ B_{M,1} & B_{M,2} & \cdots & B_{M,N} \end{bmatrix}$$

and

$$C = \begin{bmatrix} C_{1,1} & C_{1,2} & \cdots & C_{1,N} \\ C_{2,1} & C_{2,2} & \cdots & C_{2,N} \\ \vdots & \vdots & \vdots & \vdots \\ C_{N,1} & C_{N,2} & \cdots & C_{N,N} \end{bmatrix}.$$

where $n = Np = Mq$ for some integer p and q. And each $A_{I,J}$ is a $p \times q$ matrix; each $B_{I,J}$ is a $q \times p$ matrix; and each $C_{I,J}$ is a $p \times p$ matrix. It can be shown that

$$C_{I,J} = \sum_{K=1}^{M} A_{I,K} B_{K,J}. \tag{2}$$

1 Assignments

1. (20%) Prove the correctness of (2). You can reference textbook Sec 1.6.
2. (20%) Show the numbers of additions and multiplications of (1) and (2) are the same.
3. (20%) Implement the block version of matrix multiplication in C.
4. (20%) Let $n = 1024$. Try different combinations of p and q, and measure their running times.
5. (20%) Google what is *cache memory* in computer architecture, and look up the cache size of your computer. Explain the reason of the performance differences for various p and q.

2 Submission

1. Write a report in PDF file that includes the answers of question (1), (2), (4), and (5). Represent the answer of (4) in a table.

Time(seconds)	$p = 4$	$p = 8$	$p = 16$	$\cdots$	$p = 256$
$q = 4$					
$q = 8$					
$q = 16$					
$\vdots$					
$q = 256$					

2. The code of (3) should be implemented in the file `implement.c`. It should follow the interface defined in in `matmul.h`, and can be compiled together with `matmul.c`. If the code cannot be correctly compiled or run by TA, you got 0 for (3) and (4).
3. Zip them and submit.